

~~10~~ tips for writing a proposal



Jim Kurose
Department of Computer Science
University of Massachusetts

1: Pick good problem(s)

- ❑ why is the problem important?
 - ❖ what happens if you do not solve this problem?
 - ❖ why should anyone care?
- ❑ new fundamentals/principles involved?
 - ❖ universal truths for which specific problems attacked are a specific case?
- ❑ a problem area with “legs”?
 - ❖ once you’re done, is story over, or is this fundamental work leading to lots of other future work?
 - ❖ are you setting a foundation?



A fool can ask more questions in a minute than a wise man/woman (or a Yoda) can answer in a lifetime

2: Every proposal tells a story

❑ what is the “elevator pitch” of your proposal?

elevator pitch = summary short enough
to give during an elevator ride

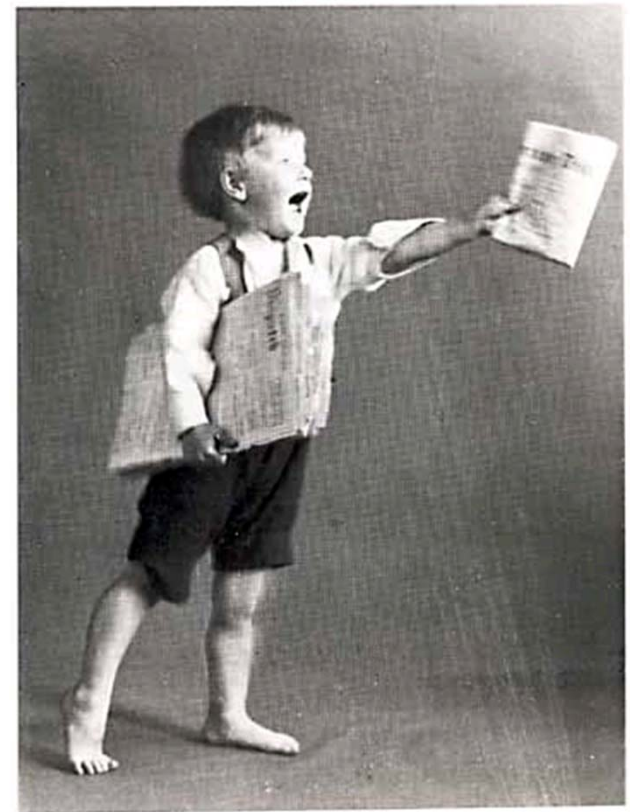
❑ story is *not* what you will do, but rather

- ❖ what you will show, new ideas,
new insights
- ❖ why interesting, important

❑ why is story of interest to others?

- ❖ universal truths, hot topic,
surprises or unexpected results

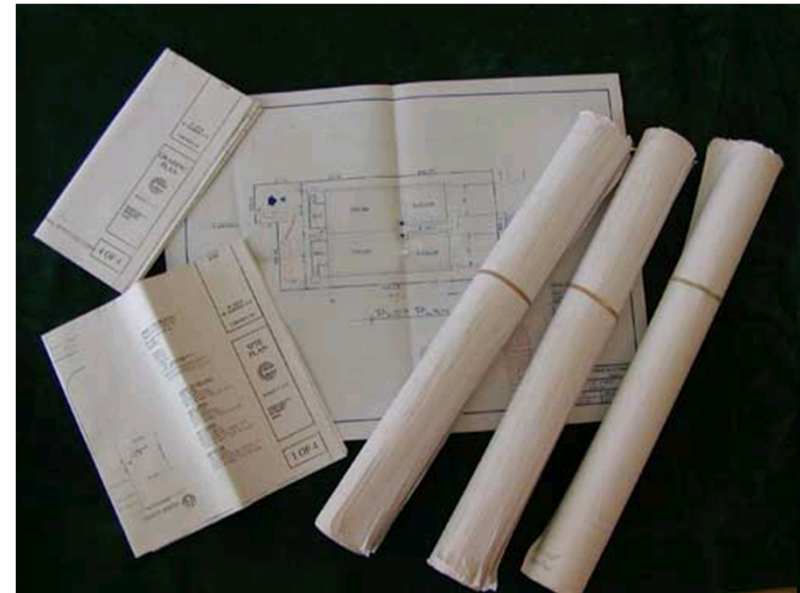
❑ know your story!



So far ... what problem area, why important?

3: **what** will you do, and **how** will you do it?

- ❑ basic questions all reviewers will ask
- ❑ so **ask and answer these questions** for the reviewers in your proposal



what – questions to be addressed

how – methodology to address questions

4: What specific research questions will you address?

- ❑ pose questions, show initial results, demonstrating methodology
- ❑ some near-term problems that you have an idea how to attack
- ❑ list longer term problems that you may only have vague idea of how to solve
 - ❖ showing longer term issues is important

5: Initial work must be done before proposal

- ❑ initial results demonstrate feasibility
 - ❖ illustrative, explanatory to reviewer
 - ❖ provide intuition about what you will do
- ❑ but not so much work that problems are basically solved already (even though they might be)
- ❑ illustrate approach(es) to solving problems
 - ❖ show you possess right skill set



6 Introduction: crucial, formulaic

- ❑ if reviewer not excited by intro, proposal is lost
- ❑ recipe:
 - ❖ *para. 1*: motivation: broadly, what is problem area, why important?
 - ❖ *para. 2*: narrow down: what is problem you specifically consider
 - ❖ *para. 3*: “In this proposal, we”: most crucial paragraph, tell your elevator pitch
 - bulleted list, **¥bf** or *¥em* initial text of what major contributions will be
 - ❖ *para. 4*: how different/better/relates to other work, at high level
 - ❖ *para 5*: summarize contributions at higher level, long-term 10K ft view of contribution: change the world!
 - ❖ *para. 6*: “The remainder of this proposal is structured as follows

7 Past work

- ❑ be specific about past related work, how proposed research differs
 - ❖ assume that reviewers are knowledgeable, aware of past work
 - ❖ what is the value added of proposed work (not just difference)



*"What Descartes did was a good step. You have added much
If I have seen a little further it is by standing on the shoulders of Giants."*

Sir Issac Newton, 1676

8. Know the review process

- ❑ reviewers may read 10-15 proposals (lots of work, tiring)
 - ❖ interesting, fun/pleasant to read proposals a rarity
- ❑ reviewers will typically be panelists present at NSF
- ❑ rank proposals and bin: must fund, fund, do not fund



9. Put yourself in place of reviewer

❑ less is more:

- ❖ “I would have sent you less if I had had time”
- ❖ *take the time to write less*

❑ reviewers shouldn't have to work

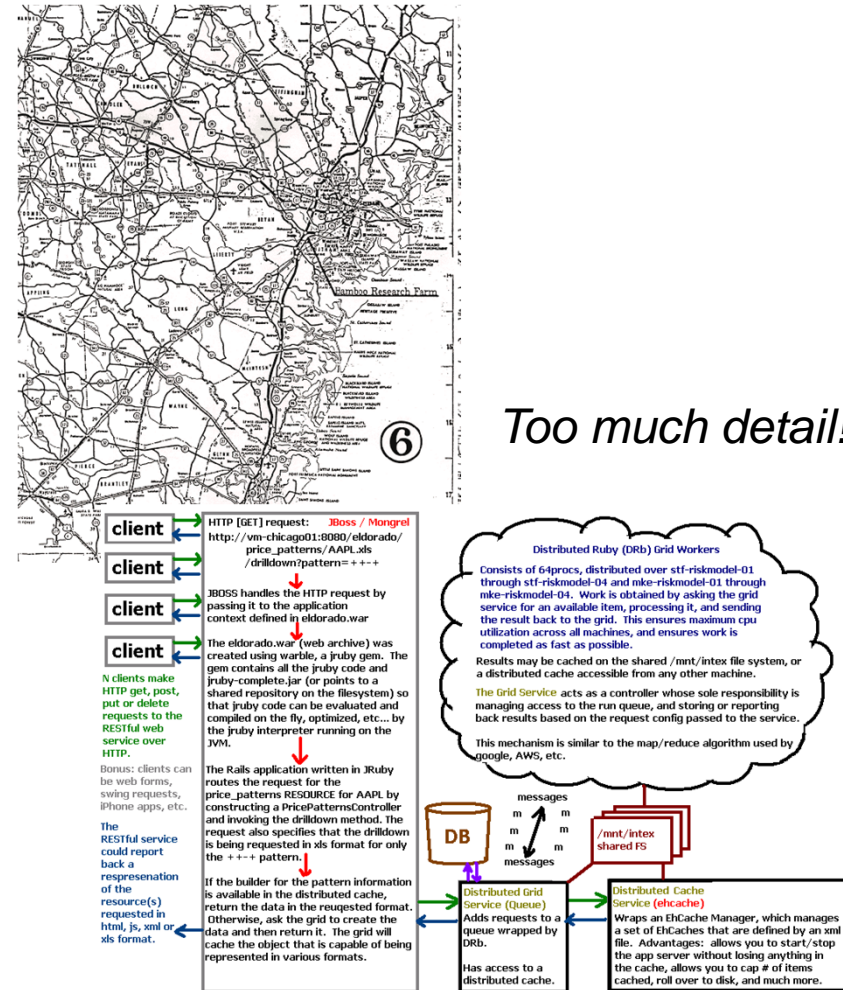
- ❖ won't “dig” to get story, understand context, results
- ❖ need textual signposts to know where ‘story’ is going, context to know where they are
 - good: “e.g., Having seen that ... let us next develop a model for Let Z be”
 - bad: “Let Z be”

❑ what does reader know/not know, want/not want?

- ❖ write for reader, not for yourself

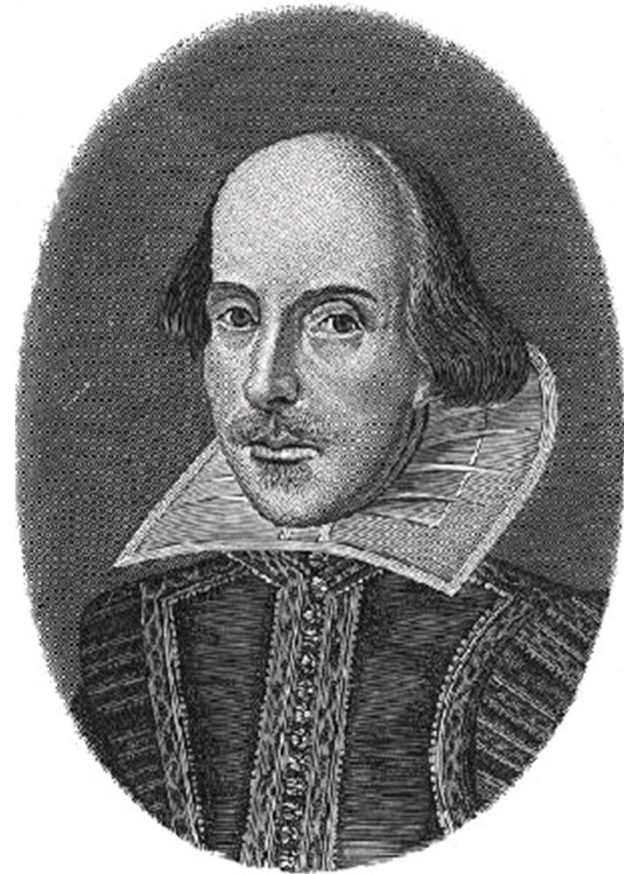
10. Put yourself in place of reviewer

- ❑ page upon page of dense text is *no fun* to read
 - ❖ avoid cramped feeling of tiny fonts, small margins
 - ❖ create openness with white space: figures, lists
- ❑ provide enough context/information for reviewer to understand what you write
 - ❖ no one has as much background/content as you
 - ❖ no one can read your mind
 - ❖ define all terms/notation



11. Master the basics of organized writing

- ❑ paragraph = ordered set of topically-related sentences
- ❑ lead sentence
 - ❖ sets context for paragraph
 - ❖ usually ties to previous paragraph
- ❑ sentences in paragraph should have logical narrative flow, relating to theme/topic
- ❑ don't mix tenses in descriptive text
- ❑ one sentence paragraph: warning!

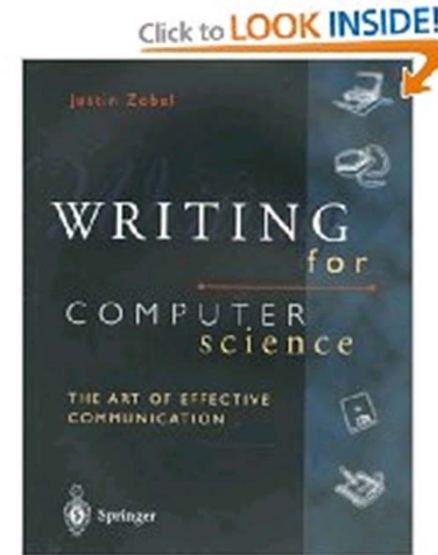


"No tale is so good that it can't be spoiled in the telling"
Proverb

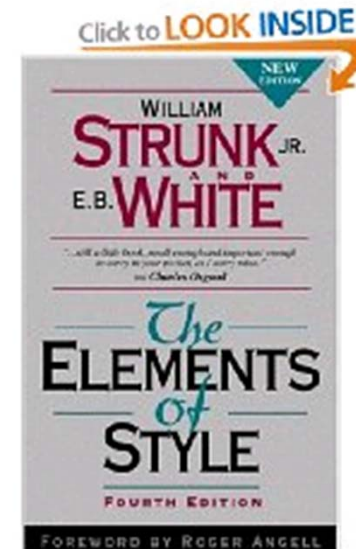
12. Write top down

- ❑ computer scientists (and most human beings) think this way!
- ❑ state broad themes/ideas/questions first, then go into detail
 - ❖ context, context, context
- ❑ even when going into detail ... write top down!

The Elements of Style
by William Strunk E. B. White
(50 years old – and still a classic!)

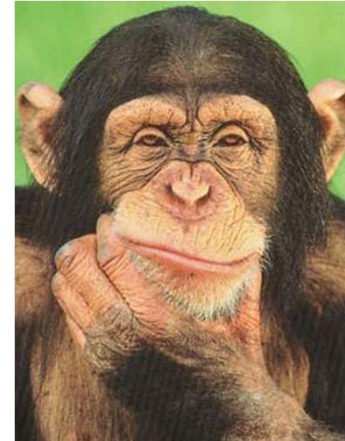


Writing for Computer Science
by Justin Zobel



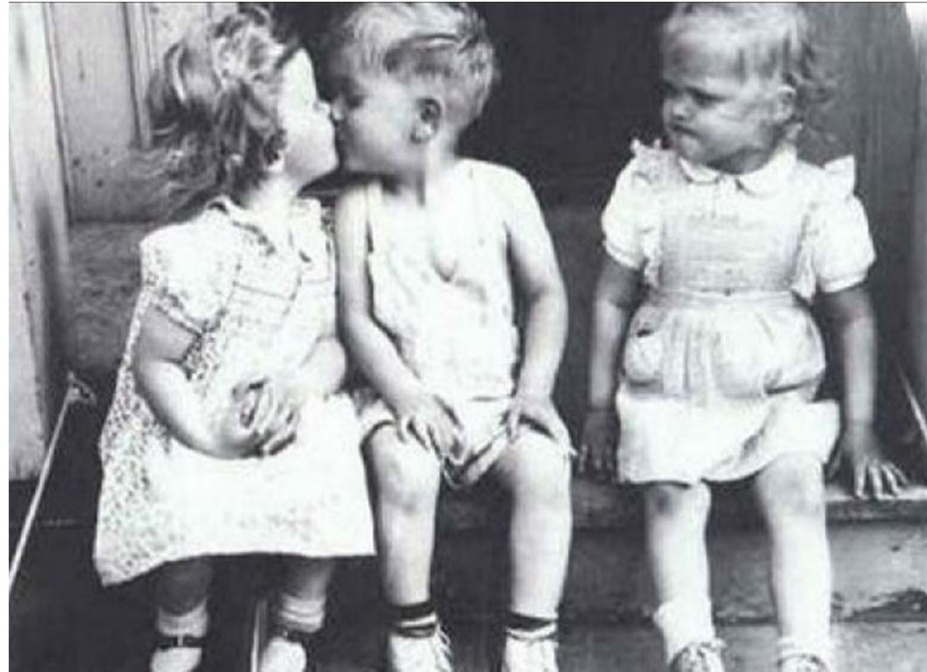
13. Good proposal writing takes time

- ❑ give yourself time to reflect, write, review, refine
- ❑ give others a chance to read/review and provide feedback
 - ❖ get a reader's point of view
 - ❖ find a good writer/editor to critique your writing
- ❑ starting a proposal two weeks before the deadline, while ideas/results are still being generated, is a non-starter
- ❑ face-to-face meetings to thrash out questions



Learn from Rejection

- ❑ it'll happen now and then, for the rest of your professional life (hopefully not with your partner)
- ❑ learn from rejection: *Why* was paper/proposal rejected? *What* did/didn't reviewers see/like?



- ❑ but don't write assuming the same reviewers will review your proposal (paper). They won't!